Towards Systematic Performance Evaluation of Non-Intrusive Load Monitoring Algorithms and Systems

Lucas Pereira and Nuno J. Nunes
Madeira Interactive Technologies Institute
Funchal, Portugal
lucas.pereira@m-iti.org, njn@uma.pt

***Abstract*—In this paper we present our approach to create an end-to-end software platform to enable the creation of meaningful and systematic, cross-dataset performance evaluations and benchmarks of Non-Intrusive Load Monitoring technology. We specifically propose a new file format to represent public datasets, a software framework to implement algorithms and metrics as well as the application of ceiling analysis to evaluate the overall performance of NILM systems.**

Keywords—NILM; performance evaluation; public datasets; file format; software framework; ceiling analysis

I. INTRODUCTION

Non-Intrusive Load Monitoring (NILM) [1] is considered, by many, one of the most promising technologies to unobtrusively identify and monitor not only the overall energy consumption, but also individual appliances that co-exist in a building, without the need to instrument every electrical device. In broad terms, it can be defined as being a set of signal-processing and machine-learning techniques used to estimate the whole-house and individual appliance electricity consumption from current and voltage measurements taken at a limited number of locations in the electric distribution of a house (optimally the mains, hence covering the demand of the entire house). Overall NILM represents a major change in the load-monitoring paradigm, posing itself as a low-cost alternative to traditional intrusive monitoring technology that involve installing sensors in each individual appliance, thus making these approaches much more expensive, intrusive and difficult to maintain.

Yet, despite all the potential and expectations of this technology to date, with the exception of a few start-ups[1], complete NILM solutions have failed to enter the market. A major obstacle to the release of finalized commercial unobtrusive energy disaggregation solutions is believed to be the almost complete lack of formal methods to evaluate the performance of the underlying algorithms, hence compromising the feasibility of the approach in real-world deployments.

Matera fact, most of NILM research to date has been conducted in very controlled laboratory settings with small and unrealistic datasets, consequently missing the complexity associated with the modern household electrical grid settings. Furthermore, there has never seem to be a consensus regarding which performance metrics should be used to measure and report accuracies, as so, despite most researchers report their results by using accuracy as the main metric, they will not always necessarily mean the same thing.

Against this background, recent years have seen the emergence of several public datasets for NILM research (e.g. REDD [2], BLUED [3], UK-Dale [4] and ECO [5]), aimed at evaluating NILM algorithms. Yet, these datasets present wide differences between them, and therefore it is still difficult (if not impossible) to produce meaningful comparative NILM evaluations across datasets. Consequently, recent times have seen some attempts to homogenize the existing datasets and provide single interfaces to run evaluations, namely the NILM metadata proposal [6], the open-source Non-Intrusive Load Monitoring Toolkit (NILMTK) [7] and the NILM-Eval framework [5].

Here we propose our approach to systematically evaluate and benchmark NILM technology across different datasets and performance metrics, using open source technologies and well-established performance metrics and evaluation techniques.

II. PROPOSED APPROACH

First, we propose the creation of a common file format and programming interface to standardize the way public datasets are stored and accessed. Second, we propose the development of an object oriented software framework to facilitate the development and implementation of NILM algorithms and metrics. And third, we propose the introduction of the *ceiling analysis* technique to evaluate the individual performance of each algorithm and how these end-up affecting the overall system performance. Next we present the rationale behind our proposals and highlight their major aspects.

A. Dataset Standardization

One of the limitations of the currently available datasets is the way these are released to the public, often requiring a great amount of work to understand the underlying structure of the data, and to produce code to interface the datasets with the different disaggregation algorithms and performance metrics.

Therefore, in the **first step** of our approach we propose the creation of a common file formant and Application Programming Interface (API) which stores and accesses public NILM datasets. Overall, our premise is that having such tools will enable the introduction of a new performance evaluation paradigm that we refer to as **"program once – evaluate everywhere"**, which contrasts greatly with the traditional approach in which researchers have to develop code to

[1] http://www.bidgley.com; http://www.plotwatt.com; http://www.yetu.com

978-3-901882-70-8 ©2015 IFIP

interface their algorithms with the different datasets, or shift to one of the programming languages used by the currently available toolkits / frameworks (e.g. NILMTK is written in Phyton and NILM-Eval was developed for MatLab).

The proposed file format (SURF) [8] is an extension of the Resource Interchange File Format - Waveform Audio File Format (RIFF-WAVE). It supports the storage of power demand data along with individual appliance and user activity information (i.e. power events and groups of power events that when combined form everyday activities, e.g. clothes washer, clothes drier and iron when taking care of the laundry), and other relevant metadata (e.g. creation date, appliances list, household details) in a single compact file.

The programming interface (SURF-PI) was implemented combining and extending several open source Java libraries for audio edition. It currently supports three main operation types: i) **Create / Update**: functions to write, edit and delete the available chunks, e.g. `WritePower(powerData);` `SetApplianceActivity(position, activityData);` ii) **Read**: functions to read chunk data, e.g. `GetFormat()`, `GetUserActivity(id);` `GetAppliances();` and iii) **Query**: functions for NILM specific queries, e.g. `GetIntervalConsumption(startPosition, endPosition);` `GetActivityConsumption(activityID).`

B. *Algorithms and Metrics Implementation*

In order to properly evaluate the different NILM algorithms it is important to be able to use the same performance metrics across the different datasets. Nevertheless, developing NILM algorithms and metrics holds its own challenges. For example, different algorithms have **different inputs** and **training methods** (e.g. event-based approaches require labeled transitions while non event-based approaches require individual appliance consumption data). Likewise, algorithms may produce **different outputs**, e.g., an event-based algorithm can produce a sequence of labeled transitions, while a non-event based algorithm may produce estimates of the energy consumed by each appliance, thus supporting the importance of having mechanisms to facilitate the process of developing such algorithms.

To this end, we propose the Open Energy Disaggregation software Framework (OpenEDF), and object-oriented software framework for developing NILM algorithms and performance metrics.

Our premise is that, if researchers are given tools tools to perform some of the most repetitive tasks (e.g. loading and writing data) they will only have to focus on the actual energy disaggregation problem rather than dealing with the low-level implementation details of providing a working system.

Fig. 1 shows a block diagram with the different software modules that currently compose our framework and how they connect to each other.

The main workflow of OpenEDF follows the dark arrows, starting from the **data acquisition** module on the left where datasets are loaded using the SURF-PI. Next, there is the option for directly computing the different power metrics in the **power calculation** module (*arrow 1*) or driving the data through the **data pre-processing** module before performing the power metrics calculation (*arrows 1a* and *1b*). Once the metrics are calculated, the following step is to proceed to the **load disaggregation** step (*arrow 2*) or optionally drive the data once again through the **data pre-processing** module (*arrow 2a*) before load disaggregation.

Regarding the load disaggregation step, OpenEDF is currently limited to event-based approaches and contains four modules for that effect.

First, the calculated power metrics are fed to the **event detection** module, which processes those values (normally the active power) to find the changes in the load that are generated by the different electric devices in the grid. Next, the **feature extraction** module further processes the detected events to extract sets of characteristics that uniquely identify each of them, which will then be used by the **event classification** module in order to find the electric device that was responsible for a given power change. Lastly, in the **energy estimation** module, the system tracks all the labeled appliance transitions and creates energy estimates for every appliance that has at least one labeled transition.

Concurrently, several performance metrics are calculated by comparing the expected results (loaded from the dataset using the SURF-PI) to those obtained in the load disaggregation steps. These metrics include traditional statistics for edge detection and classification problems (e.g. *Precision*, *Sensitivity*, *F-1 Score* and the *Receiver Operating Characteristics*) as well as several energy estimation metrics such as the *total energy explained*, the *energy identification rate* and the *estimated and true power difference*.

Lastly, following the light arrows, one can see how the main workflow modules can communicate with utility modules such as **storage**, **visualization** and **data transmission**.

C. *Performance Evaluation*

Overall, the NILM performance evaluation to date focuses in the performance of individual algorithms, e.g. event detection and event classification for event-based approaches, or dynamic and constraint programming algorithms in the case of non event-based approaches.

Yet, NILM solutions contain different algorithms, and since the performance of a particular algorithm may depend on that of its predecessor, the individual performance of each one may not enough to assess the overall system performance. Furthermore, some datasets contain considerably more data than others, which in the one hand can result in better performances, but on the other hand may require longer training. Thus introducing a trade-off between disaggregation and execution performance that also needs exploring by the research community.

Consequently, we believe that in order to properly assess the quality of NILM systems a new kind of tool that enables case-by-case reasoning is required. To this end, we propose the introduction of the *ceiling analysis* technique to the problem of evaluating NILM algorithms and systems.

Originally developed to help machine-learning practitioners understand how each step in the pipeline would positively or negatively impact the final results, *ceiling analysis* work by estimating how the final error is affected by each stage of the pipeline by substituting labeled data in the stage under evaluation, then revealing how well the system would work if that stage had no error (or a pre-established error). Stepping through each of the pipeline stages will therefore highlight each stage's potential to improve or degrade the overall system performance.

In particular, we wish to explore the potential of applying *ceiling analysis* in NILM systems such that it is possible to understand the importance of each algorithm in the NILM machine-learning pipelines by quantifying the effects of the possible errors (e.g. missing a power change, detecting a wrong power change, a wrong classification, etc.), and also answer some important open research questions such as: i) how much training data do we need to consistently achieve good results? and ii) what are the effects of unlabeled or poorly labeled data?

To this end we propose the development of a software toolkit for performing *ceiling analysis* of NILM systems on top of the OpenEDF software framework, just described.

By following this approach we expect to gain a deeper understanding about the different data requirements (datasets, file formats and programming interfaces) for each approach, the different algorithms and which performance metrics are best suited for evaluating each one of them.

III. Conclusion and Future Directions

In this paper we presented our approach to produce systematic evaluations of NILM algorithms and systems.

At the time of this writing we have managed to successfully convert the one-week long BLUED dataset to our format and perform in-depth evaluations of different event detection algorithms. Since BLUED contains only one week of data we couldn't perform any meaningful evaluation of event classification algorithms. We are currently in the process of converting the UK-Dale dataset, which contains several months of data for 5 households in the UK, such that it is also possible to evaluate event classification algorithms using our proposed approach. To this end, and considering the size of the UK-Dale dataset we developed a semi-automatic labeling tool as described in [9].

Lastly, SURF, SURF-PI and OpenEDF are all open-source. Early prototypes are available for other researchers to explore and download from the *Sustainability for Smart Cities* research group website at *http://s4sc.m-iti.org*.

References

[1] G. W. Hart, "Prototype Nonintrusive Appliance Load Monitor," MIT *Energy Laboratory Technical Report, and Electric Power Research Institute Technical Report*, Sep. 1985.

[2] Z. Kolter and J. Matthew, "REDD: A public data set for energy disaggregation research," presented at the *Workshop on Data Mining Applications in Sustainability (SustKDD)*, San Diego, CA, USA, 2011.

[3] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Berges, "BLUED: A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research," *presented at the Workshop on Data Mining Applications in Sustainability (SustKDD)*, Beijing, China, 2012.

[4] J. Kelly and W. Knottenbelt, "'UK-DALE': A dataset recording UK Domestic Appliance-Level Electricity demand and whole-house demand," in *arXiv:1404.0284 [cs]*, 2014.

[5] C. Beckel, K. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, "The ECO Data Set and the Performance of Non-Intrusive Load Monitoring Algorithms," in *1st ACM International Conference on Embedded Systems for Energy-Efficient Buildings*, Memphis, TN. USA, 2014.

[6] J. Kelly and W. Knottenbelt, "Metadata for Energy Disaggregation," *ArXiv14035946 Cs*, Mar. 2014.

[7] N. Batra, J. Kelly, and O. Parson, "NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring," in *Proceedings of the 5th International Conference on Future Energy Systems*, Cambridge, UK, 2014.

[8] L. Pereira, N. Nunes, and M. Bergés, "SURF and SURF-PI: A File Format and API for Non-Intrusive Load Monitoring Datasets," in *Proceedings of the 5th International Conference on Future Energy Systems*, Cambridge, UK, 2014.

[9] L. Pereira and N. J. Nunes, "Semi-Automatic Labeling for Public Non-Intrusive Load Monitoring Datasets," in *Proceedings of the 4th IFIP/IEEE Conference on Sustainable Internet and ICT for Sustainability*. Madrid, Spain

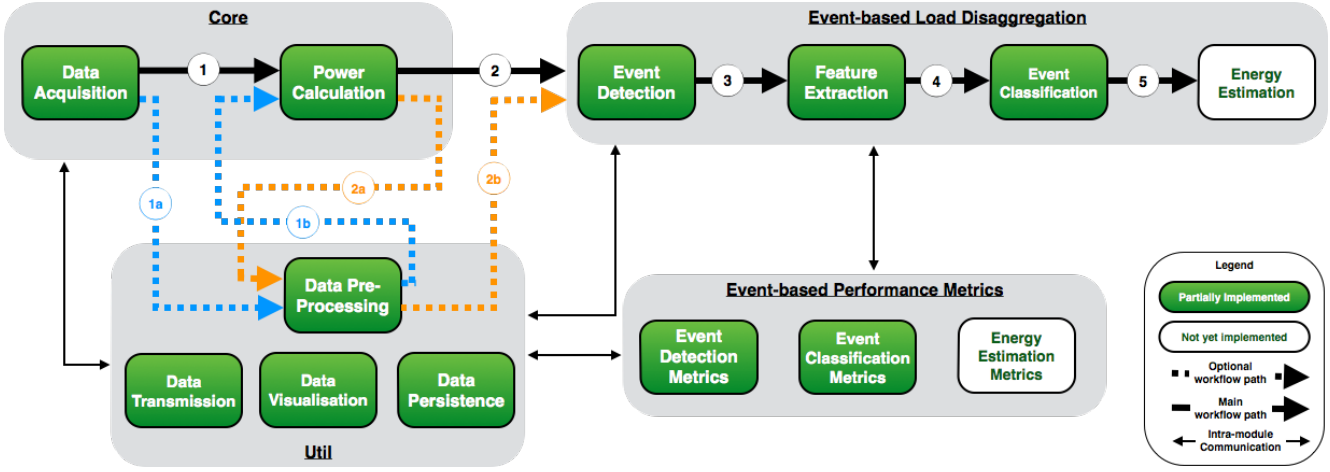

Fig. 1. Open Energy Disaggregation software Framework block diagram

FCT (SFRH/BD/77856/2011)
LARSyS (PEst-OE/EEI/LA0009/2015)